# A Tool for Visualizing the Topology of Three-Dimensional Vector Fields

A. Globus, Computer Sciences Corporation[1]
C. Levit, NASA Ames Research Center
T. Lasinski, NASA Ames Research Center

## Abstract

*We describe a software system, TOPO, that numerically analyzes and graphically displays topological aspects of a three dimensional vector field,* **v***, to produce a single, relatively simple picture that characterizes* **v***. The topology of* **v** *that we consider consists of its critical points (where* $\mathbf{v} = 0$*), their invariant manifolds, and the integral curves connecting these invariant manifolds. Many of the interesting features of* **v** *are associated with its critical points. The field in the neighborhood of each critical point is approximated by the Taylor expansion. The coefficients of the first non-zero term of the Taylor expansion around a critical point are the 3x3 matrix* $\nabla\mathbf{v}$*. Critical points are classified by examining* $\nabla\mathbf{v}$*'s eigenvalues. The eigenvectors of* $\nabla\mathbf{v}$ *span the invariant manifolds of the linearized field around a critical point. Curves integrated from initial points on the eigenvectors a small distance from a critical point connect with other critical points (or the boundary) to complete the topology. In addition, one class of critical surfaces important in computational fluid dynamics is analyzed.*

*TOPO is implemented as a module in the FAST [1] visualization environment. FAST is general purpose visualization software with modules for isosurface generation, particle tracing, etc. TOPO operates on curvilinear, structured grids, including large multi-zone grids. We have used TOPO to visualize a number of computational fluid dynamics (CFD) data sets. The results agree well with other topology software and hand generated topologies. TOPO has proved useful in finding surface topology, flow attachment and separation points, vortex cores, scalar field local extrema, and generally interesting regions of* **v***. We believe there may be other interesting applications yet to be discovered. This paper, along with the references, contains most of the information needed for a scientific programmer to code a topology module in another environment.*

## Introduction

Three dimensional vector fields are difficult to visualize. One simple technique chooses a set of points in the field and draws arrows indicating the magnitude and direction of **v** at each point. Unfortunately, this usually results in a display that is either cluttered or limited to a small subset of the data. If the points are chosen from some simple shape like a plane, the ends of the vectors can be connected to form a deformation surface -- but again only a subset of data is visualized. One can calculate the vector magnitude and use scalar visualization techniques but directional information is lost. Interactive control of initial positions for integral curves (particle traces) [2] and surfaces [3] may be used to explore **v**, but choosing appropriate initial positions is hardly straightforward.

A simple picture completely characterizing **v** is the ideal. Studying such a picture should give one a clear and complete understanding of the important characteristics of **v**. Vector field topology visualization can go a long way in this direction; our software takes one small step.

Informally, vector field topology consists of the key points, curves and surfaces that, taken together, characterize all integral manifolds in **v**. Integral manifolds include particle traces, streamlines, integral curves, stream surfaces and integral surfaces. With certain exceptions, all integral manifolds must begin and end at zeros in **v** -- or at field boundaries. The zeros in **v** form points, curves, surfaces and volumes -- critical manifolds. Thus, if one finds the critical manifolds in **v** and can characterize **v** in the neighborhood of these manifolds, one can generate almost any integral manifold in **v** by choosing initial conditions according to the characterization.

## Previous Work

The connections between differential equations and the topology of vector fields were initially developed by Poincare, who laid down the foundations of the qualitative theory of differential equations in 1875 [4]. Andronov essentially completed the theory in two dimensions during the middle of the twentieth century [5,6]. The theory in three dimensions is still not complete [7]. More contemporary tutorial references which discuss differential equations, vector spaces and dynamical systems include [8,9,10]. Since the introduction of computers, a plethora of complex phenomena that require at least three dimensions to exist (e.g., chaotic flows and strange attractors) have become much more widely known. Many of them were, however, known to Poincare. For a historical survey, see [11].

One of the first applications of the qualitative theory of differential equations to three dimensional fluid flows was by Lighthill [12]. Perry and Fairlie [13] extended the approach to a more general class of flows. This initial work has been enhanced by that of many others, notably [14,15,16,17].

Automatic computation of fluid flow topology from ex-

[1] Work supported under NA2-12961.

perimentally generated data has been investigated [18]. The topology of numerically simulated flowfields has been investigated using automatic techniques by Shirayama and Kuwahara [19] and by Helman and Hesselink [20,21,22,23].

Helman and Hesselink have reported on two dimensional topology software using a critical point classification scheme similar to ours. A third dimension is used to represent time for unsteady 2D flows. They create surfaces connecting the important integral curves at different time steps. They also find attachment and separation surfaces in three dimensional flows. Their two dimensional results on the hemisphere cylinder surface flow topology agree very well with ours. Recently, their work has been extended to find 3D critical points [22].

## Analysis

TOPO analyzes data generated by computational fluid dynamics (CFD) codes. These data are expressed on curvilinear, structured grids in PLOT3D format [26]. Trilinear interpolation is used to find field values between grid points. The curvilinear coordinate system is often referred to as "computational space" in this paper. The curvilinear coordinates of a point are $\xi_i = (\xi, \eta, \zeta)$ while its physical coordinates are $x_i = (x,y,z)$. The two coordinate systems are related by the transformation:

$$\xi_i = \xi_i(x_i) \qquad x_i = x_i(\xi_i) \tag{1}$$

This coordinate system is usually implemented as a three dimensional array of (x,y,z) positions. The i,j,k indexes into this array are equivalent to the integer part of the $(\xi, \eta, \zeta)$ coordinates. We use computational space for calculations since they simplify many operations; e.g., differencing.

$\mathbf{v} = (u, v, w)$ is defined as:

$$v_i = \frac{dx_i}{dt} \tag{2}$$

where, in this case, (x,y,z) are the lagrangian coordinates of an element "moving" with the vector field and t is time in velocity fields and an arbitrary parameter in others.$\mathbf{v}(x, y, z)$ may be converted to computational space $\tilde{\mathbf{v}}(\xi, \eta, \zeta)$ using the chain rule. Considering only the first component:

$$\tilde{u}(\xi, \eta, \zeta) = \frac{d\xi}{dt} = \frac{\partial\xi}{\partial x}\frac{dx}{dt} + \frac{\partial\xi}{\partial y}\frac{dy}{dt} + \frac{\partial\xi}{\partial z}\frac{dz}{dt}$$

or using matrix vector notation to include all components,

$$\tilde{\mathbf{v}} = \mathbf{J}^{-1}\mathbf{v} \quad , \qquad \mathbf{J}_{ij} = \frac{\partial x_i}{\partial \xi_i} \tag{3}$$

where **J** is the Jacobian matrix of the coordinate transformation. **J** and $\mathbf{J}^{-1}$ are used to convert between physical space and computational space.

Critical points exist where the components of **v** all simultaneously vanish. They are identically located in both physical and computational space since zero length vectors remain invariant under nonsingular local linear transformations such as **J** and $\mathbf{J}^{-1}$.

**The Matrix $\nabla\mathbf{v}$:** The Taylor series expansion of **v** about a point $x^{(0)}$ is:

$$v_i = v_i^{(0)} + (x_j - x_j^{(0)})\frac{\partial v_i}{\partial x_j} + O(\Delta x_k \Delta x_l) \tag{4}$$

we assume **v** sufficiently smooth and differentiable for the Taylor expansion to exist. Subscripts indicate components. At a critical point, the first term of the expansion vanishes (by definition). Considering only the second term, each equation has three terms, one for each coordinate direction. The coefficients of these equations are the 3x3 matrix $\nabla\mathbf{v}$.

$$(\nabla\mathbf{v})_{ij} = \frac{\partial \mathbf{v}_i}{\partial \mathbf{x}_j} \tag{5}$$

Around a critical point, the eigenvalues and eigenvectors of this matrix determine the local behavior **v**. Positive eigenvalues indicate that **v** is directed away from the critical point (a repelling eigendirection) and negative values the opposite (an attracting eigendirection). A complex conjugant pair of eigenvalues indicate that **v** spirals in or out, depending on the sign of the real part of the eigenvalues. Thus the linear approximation of **v** near a critical point is characterized by the eigenvalues and eigenvectors of $\nabla\mathbf{v}$. The eigenvalues of **v** may be used to classify the type of a critical point and the eigenvectors may be used to find its invariant manifolds.

As shown in reference [37] the matrix $\nabla\mathbf{v}$ is related by a similarity transformation to its computational space equivalent $\nabla\tilde{\mathbf{v}}$. Thus the eigenvalues of both matrixes are the same, and eigenvectors are related by the same similarity transformation.

**Critical Point Classification:** Critical points may be classified as nodes, foci (2D only), or saddles on the basis of the eigenvalues of $\nabla\mathbf{v}$ [11]. Nodes and foci may be further classified as attracting or repelling (See Figures 1 and 2). Positive eigenvalues indicate a repelling node, negative values an attracting node.

Two dimensional saddles have one positive and one negative eigenvalue. Near a saddle, **v** approaches the critical point along negative eigendirections and recedes along positive eigendirections. In three dimensions, two eigendirections have the same sign and span a plane. The third eigendirection spans a line. Thus, **v** approaches (for example) a 3D saddle along a plane and recedes (in this example) from the saddle along a line.

Foci only appear in two dimensions. Around foci, **v** spirals towards or away from the focus. The eigenvalues are a complex conjugant pair with a positive real part indicating a repellor and a negative real part indicating an attractor. The magnitude of the imaginary part indicates the strength of the spiraling motion. In three dimensions, the plane of a saddle may be a two dimensional focus. Such a 3D saddle will have one real and two complex eigenvalues.

The following diagrams of critical point types are after Abraham [11].

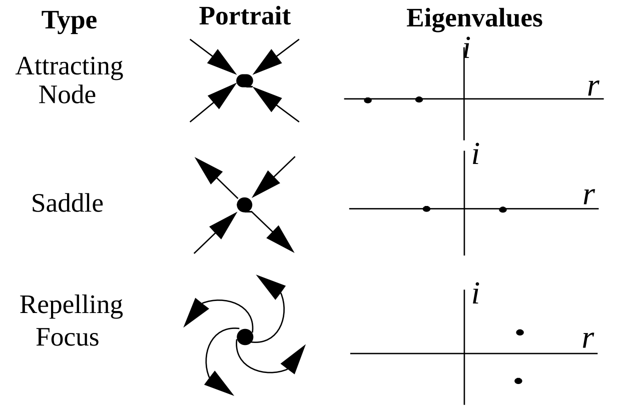


Figure 1: Classification of two dimensional critical points.

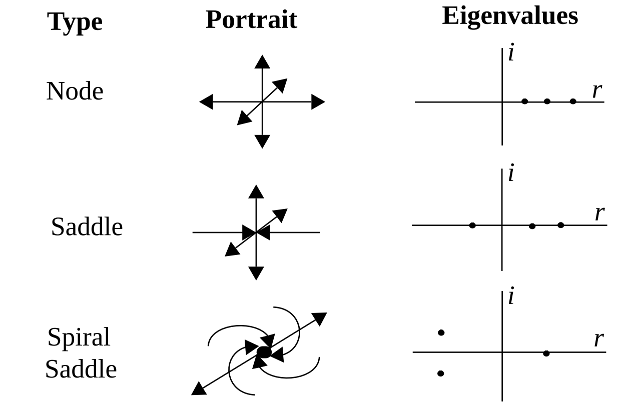


Figure 2: Classification of three dimensional critical points.

## Visualization

Our software draws the topology of **v** as a set of critical points and associated integral curves. The user selects which critical points are to be displayed based on their types. Integral curves are chosen by selecting the eigendirections to integrate along for each type of critical point.

Critical points are visualized using glyphs consisting of three lines crossing at the location of the critical point (see Figure 3). The lines are colored to distinguish the attracting and repelling eigenvectors (associated with negative and positive eigenvalues respectively). Arrowheads may be added to the lines to further clarify their direction. Lines representing vectors with complex eigenvalues are connected to form a rectangle in the invariant (spiraling) plane. Line length may be scaled by the eigenvalue's real part. One may request quantitative information on selected critical points, such as their location, eigenvalues, eigenvectors and the components of $\nabla\mathbf{v}$. Most visualization parameters are user controlled. See the reference [37] for defaults.

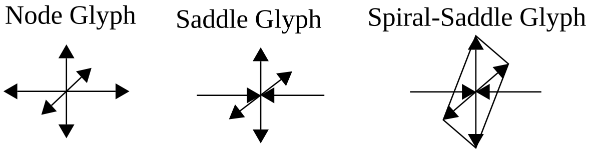


Figure 3: Critical point glyphs.

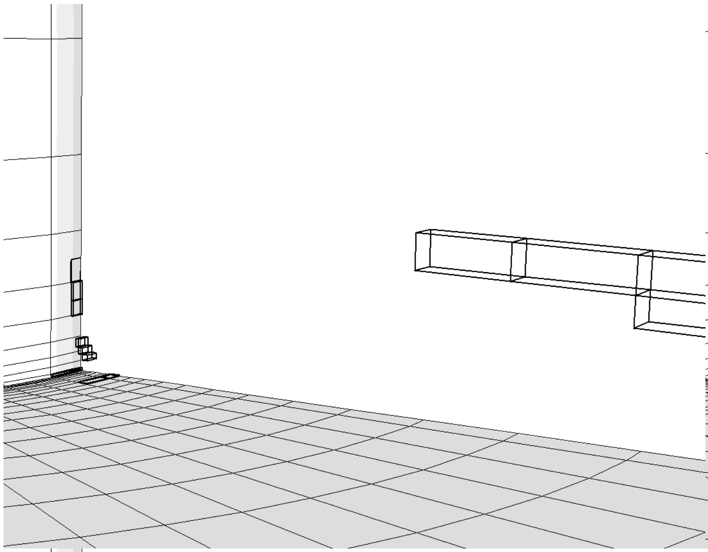
Integral curves are visualized by a set of connected line segments. Separate color controls for outgoing and incoming integral curves are provided. Optionally, arrowheads showing the direction of **v** along an integral curve may be added. These arrowheads may be animated to appear to flow along the curves.

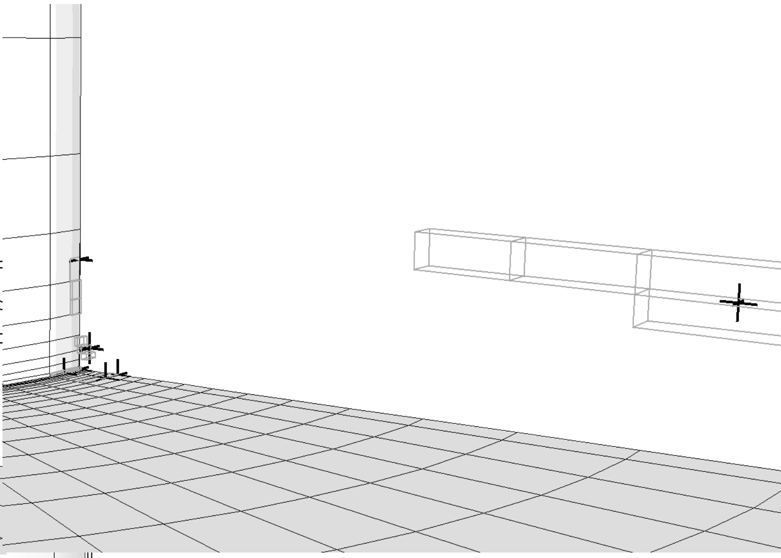
## Implementation

TOPO is implemented as a module in the FAST CFD visualization environment [2]. The software transforms **v** to computational space, locates candidate grid cells that may contain critical points, finds critical points within these candidate cells, classifies each critical point using the eigenvalues of $\nabla\mathbf{v}$, integrates curves along the eigendirections of $\nabla\mathbf{v}$, and displays critical points and integral curves combined with other FAST generated visual elements. Most parameters of topology generation are under user control. We have chosen defaults for these parameters that work well with the CFD data sets we have examined. These defaults may be found in reference [37].

**Transformation of v to Computational Space:** The vector field is converted to computational space. This transformation is accomplished by differencing **x** with respect to $\vec{\xi}$ to generate **J** at each grid node, inverting **J**, and transforming **v** using the resulting matrix. All of the following calculations are accomplished in computational space.

**Finding Candidate Grid Cells:** A critical point can only occur in a cell where the values of all three components of **v** pass through zero (see Figure 4). For monotonic interpolation schemes (e.g. trilinear interpolation), this may be determined by a simple heuristic. For each component we examine the value at each cell vertex. If both negative and positive values exist, that component must change sign (and hence pass through zero) somewhere within the cell. This is a necessary, but not sufficient condition for a critical point to exist within the cell. It is not sufficient because the surfaces within a cell where the component-wise zero cross-





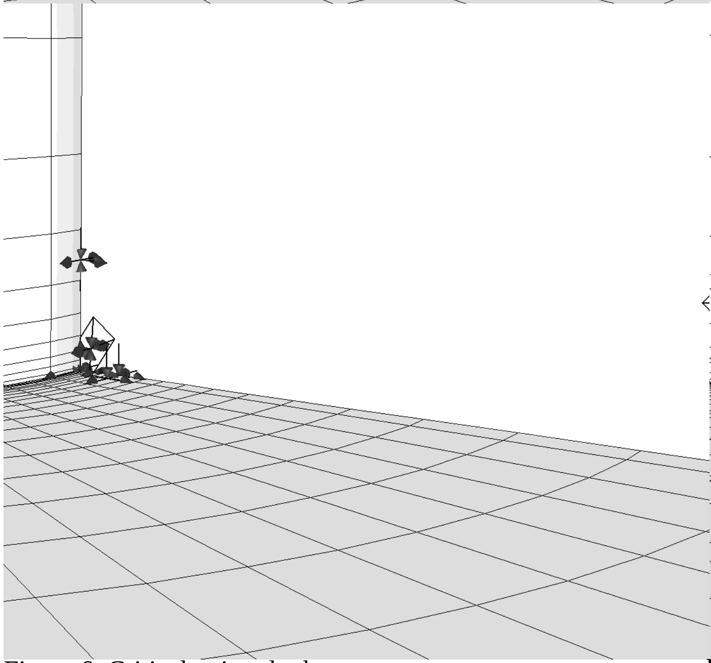


Figure 6: Critical point glyphs.

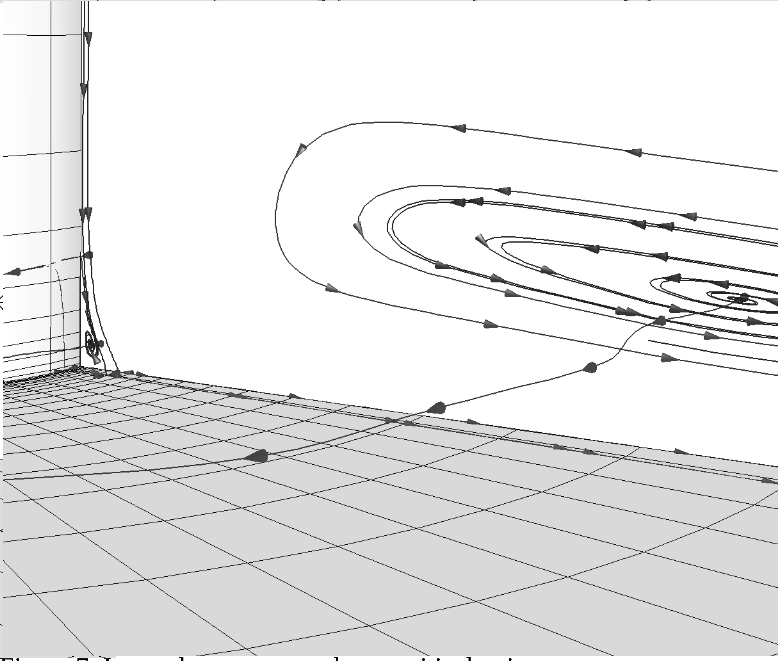


Figure 7: Integral curves started near critical points.

ings exist might not intersect.

**Finding Critical Points within Candidate Cells:** The problem of finding critical point positions inside candidate cells is equivalent to solving a system of simultaneous nonlinear equations.

The candidate cell is recursively bisected and each of the resulting eight sub-cells subjected to the candidate test. If, after a fixed number of bisections a subcell passes the candidate test, the location is more precisely estimated using Newton's method. The final point is allowed to lie within a small distance outside of the subcell. The convergence criteria for the Newton iterations is determined by multiplying the average magnitude of the cell's vectors by a small factor. This can sometimes cause the criteria to be too small, so there is also an absolute lower bound on the criteria.

The user may choose to use only the bisection method. In this case a critical point is assumed to lie at the center of any subcell passing the candidate test after the last bisection has been accomplished.

Critical point locations are shown in figure 5.

**Classifying Critical Points:** Once a critical point is located, it must be classified by examining $\nabla\mathbf{v}$. $\nabla\mathbf{v}$is calculated using finite differences to find the gradient at the cell's vertices. $\nabla\mathbf{v}$ at the critical point is then found by trilinear interpolation. This is algebraically equivalent to interpolating stencil values in neighboring cells and then differencing. The eigenvalues of $\nabla\mathbf{v}$ are found (using standard methods) and the critical point is classified based on their locations in the complex plane (see Figure 1, 2, 3, and 6).

**Integrating Curves:** Integral curves may be computed from initial positions very close to critical points along the eigendirections of $\nabla\mathbf{v}$ (see Figure 7). These curves are integrated forward or backward depending on the sign of the eigenvalue.

When an integral curve crosses a computational space boundary, the rest of computational space must be checked to see if the curve re-enters. For example, the grid may be periodic, or may overlap another grid in a multiple-zone dataset. The check is accomplished by an octree search [24] using bounding boxes to prune the search tree followed by a tetrahedral containment test [25]. A bisection method is applied to remaining potential grid cell locations before final rejection or acceptance of a new computational space position for continuing the integration. The computational space coordinate of grid re-entry is set to the center of the enclosing subcell.

**Displaying Topology:** Critical point and integral curve locations are converted from computational space to physical space for display purposes. The mapping is accomplished using trilinear interpolation of the physical space coordinates at the vertices of grid cells.

## Numerical Methods

Care must be taken when evaluating the eigensystems of matrices that are defective or nearly so [28]. Numerical ambiguities that arise when at or near a multiple root are handled in an ad-hoc manner. Results of subtraction are set to zero if the absolute value of the result is less than a small fraction of the absolute value of any operand.

In most places where the software tests a value for zero there is a relative fuzz factor. Values with an absolute value less than the relative fuzz factor are assumed to equal zero. Default fuzz factors may be found in reference [37].

Integration is performed in computational space using a fourth order Runga-Kutta method with adaptive step size error control [29]. We found this to be necessary since the magnitude and direction of the vector field can vary dramatically near critical points. The initial step size, maximum number of steps, maximum stepsize, minimum and maximum step length, and the adaptation criteria in each dimension parameterize the integration.

Differencing is accomplished using a three point stencil where possible. Where one of the values is unavailable due grid boundaries or invalid grid points, a two point, one sided difference is used.

Cell bisection is accomplished by interpolating the values at the midpoint of each cell edge, face, and at the cell center, forming eight sub-cells. This procedure is performed recursively.

## Special Cases

**Degenerate Critical Points:** The discussion thus far has focused on “generic” or “hyperbolic” critical points. Exceptional cases can arise in several situations. Non-hyperbolic critical points occur when the real part of any eigenvalue is equal to zero. Other exceptional cases occur when defective matrices are encountered and hence eigenvectors coincide. These degenerate cases, though unstable, do occur in flows with imposed constraints such as symmetry or incompressibility. Currently, all degenerate critical points are placed in a single class and no further analysis is attempted.

**Critical Curves, Surfaces and Volumes:** With the notable exception of no-slip boundaries in CFD velocity fields, TOPO makes a minimal effort to detect, analyze, and display curves, surfaces or volumes where **v** vanishes. No attempt is made to find critical curves or surfaces unless they fall on grid cell boundaries. If two adjacent grid points have zero length vectors, we assume a critical line between them. For the most part, TOPO simply draws a line between such points. Critical surfaces on cell boundaries can be detected by examining these lines. Critical volumes can be similarly found. The cells containing such critical lines are not considered candidates and are not searched for critical points.

If a critical curve or surface exists within a grid cell, then multiple critical points may be found. TOPO limits the total number of critical points found in a single grid cell to about eight. A warning message is printed if this limit is exceeded.

**No-slip Boundaries:** In many CFD computations, no-slip boundary conditions are imposed on the velocity field. On these boundaries, **v** is zero. Our software analyzes this important class of degenerate critical surface by examining the skin friction field. This is the two dimensional field formed by taking the limit of **v** at the no-slip surface. This is approximated by the value of **v** one grid line away from the surface. The critical points in the skin friction field are found and analyzed as follows: Since $\nabla \mathbf{v} = 0$ at critical points in this field, the second derivative term in the Taylor expansion of **v** is examined. This third rank tensor is identically zero in the two dimensions corresponding to the local on-surface coordinates. Thus, its non-trivial terms can be expressed as a 3x3 matrix, and the same eigenvalue analysis may be applied. This will be elaborated upon in a future paper.

To allow for comparison with topologies derived from wind tunnel oil flow experiments, integral curves starting along eigenvectors that lie on a no-slip surface are integrated in the skin friction field (i.e., the limit of **v** as the surface is approached) and constrained to stay on a no-slip surface.

**Grid Singularities:** Special case code has been included to handle common coordinate system singularities. A problem occurs in candidate grid cell selection when a grid cell edge is collapsed; i.e., if two or more adjacent grid points have identical locations in physical space. If the candidate test is applied using the computational space vectors, false positives can result. Subsequently, a critical point that does not really exist may be found. This problem can be circumvented by detecting the case where two grid points are identically located in physical space and arbitrarily choosing one of the computation space vectors to be the vector at both grid points. The candidate test will then work properly.

## Results

We have used TOPO and other FAST modules to investigate the properties of many CFD solutions.These solutions include flow about a blunt fin [30], a NASA space shuttle orbiter [31], a shuttle engine liquid oxygen post [32], a shuttle launch configuration [33], a hemisphere cylinder [34], and others.

By carefully choosing the critical points to display and the eigendirections to integrate, one may create specific visualizations. Useful visualizations include vehicle surface topology, vortex cores, scalar field extrema, interesting regions of **v**, and places where CFD flow solvers have not performed well.

**Surface Flow Topology:** By integrating curves in the two dimensional skin friction field along the eigendirections of saddles, one may visualize surface flow topology [12,21]. These curves connect skin friction field critical points and allow the user to deduce the rest of the surface skin friction field's dynamics qualitatively, since integral curves may not intersect, except at critical points. Furthermore, by examining the off-surface eigenvectors, one may determine if the flow is attaching or separating.

TOPO has been partially validated by comparing surface flow topology results on a hemisphere cylinder with Helman's [21] computed results and hand generated results in [34]. Note the close correspondence between Helman's results (Plate 1a) and ours (Plate 1b). In reference [21] Helman's results are shown to compare well with Ying's hand generated results [34].

**Vortex Cores:** By one definition, a vortex core is the integral curve within a vortex that has minimum curvature [35]. If there is a critical point on a vortex core, then that point must be a spiral-saddle [36]. The eigenvector belonging to the only real eigenvalue of the spiral-saddle corresponds, locally, to an integral curve entering or leaving the critical point. This particular curve does not spiral at all and is therefore on the vortex core. By continuing to integrate this curve, the entire vortex core may be visualized.

A shuttle main engine liquid oxygen post data set is used to illustrate TOPO's vortex core capability (see Plate 2a). The core location closely corresponds to the vortex cores found using interactive particle tracers (see Plate 2b).

**Scalar Field Extrema:** After taking the gradient derivative of a scalar field one may visualize the topology of the resulting vector field. The nodes of this field will be at scalar field local minima and maxima. Other aspects of gradient derivative fields should shed light on the structure of the corresponding scalar fields as well. We have not yet analyzed these.

**Interesting Regions of v:** Interesting regions of **v** tend to contain critical points. For example, flow reversal (recirculation) frequently involves the simultaneous existence of several critical points. One may therefore use critical point locations as a guide for the interactive specification of initial positions for particle traces and stream surfaces. For example, a cluster of critical points is found near the lee surface of the hemisphere cylinder. These critical points are closely associated with the separation bubble (see Plate 3).

**Where a Flow Solution is Inadequate:** The existence of pure attractors or pure repellors in a steady or incompressible flow solution may well indicate some problem, such as an incompletely converged solution. Such a situation cannot exist in a physically realistic flowfield since it violates conservation of mass expressed by the continuity equation $\nabla \bullet \mathbf{v} = 0$. Similarly, the existence of saddle-node pairs that disappear under a small perturbation may indicate "wiggles" in the topology, akin to the nonphysical wiggles found around shocks when inadequate differencing schemes are employed.

**Large Data Sets:** Since the topology of **v** must only be calculated once, topological visualization is an excellent technique for examining large data sets where slow response renders interactive techniques ineffective. Plate 4 are the vortex cores of a shuttle launch configuration. This is a 900,000 node, nine grid data set.

## Summary

Vector fields may be visualized using the new TOPO module in FAST. A single, relatively simple picture captures many key features of **v**. We have used this software to visualize CFD solutions. To our knowledge, the software is unique in displaying off surface eigenvectors of skin friction critical points, integrating vortex cores from spiral-saddles, and in the combination of topology visualization with a general purpose visualization system (FAST). We believe that topology visualization may be useful well beyond the grounds we have explored.

## Acknowledgments

We thank P. Buning for numerous informative discussions regarding this work.

## References

[1] G. Bancroft, F. Merritt, T. Plessel, P. Kelaita, R. McCabe, A. Globus, "FAST: A Multi-Processing Environment for Visualization of CFD," *Proc.Visualization '90*, IEEE Computer Society, San Francisco (1990).

[2] C. Levit and S. Bryson, "A Virtual Environment for Exploration of Three Dimensional Flowfields," SPIE paper 1457-19 *SPIE Conf on Stereoscopic Displays and Applications II*. San Jose (1991).

[3] J.P.M. Hultquist, "Interactive Numerical Flow Visualization Using Stream Surfaces," *Computing Systems in Engineering* **1** (2-4) pp. 349-353.

[4] H. Poincare, Sur les courbes defines par une equation

differentielle. *J. Math.* **1**, 167-244 (1875); **2**, 151-217 (1876); **7**, 375-422 (1881); **8**, 251-296 (1882); “Les Methodes nouvelles de la mecanique celeste” Vol 1. Gauthier-Villars, Paris, 1892; Paris, 1928. “OEuvres,” Vol. 1 Gauthier-Villars.

[5] A.A. Andronov, E.A. Leontovich, I.I. Gordon, and A.G. Maier, “Qualitative Theory of Second-Order Dynamic Systems,” Translated from Russian by D. Louvish, Halsted Press, John Wiley and Sons, New York, Toronto; Israel Program for Scientific Translations, Jerusalem, London.

[6] A.A. Andronov, E.A. Leontovich, I.I. Gordon, and A.G. Maier, “Theory of Bifurcations of Dynamic Systems on a Plane (Teoriya bifurkatsii dinamicheskikh sistem na ploskosti),” Izdatel’stovo “Nauka”, Glavnaya Redaktsiya, Fizido-Matematicheskoi Literatury, Moskva (1967). Translated from Russian, Israel Program for Scientific Translations, Jerusalem (1971).

[7] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields,* Springer Verlag (1983).

[8] M. Braun, *Differential Equations and Their Applications*, Springer-Verlag, New York (1978).

[9] M. Hirsch and S. Smale, *Differential Equations, Dynamical Systems and Linear Algebra*, Academic Press, New York (1974).

[10] V.I. Arnold, *Ordinary Differential Equations*, MIT Press, (1973).

[11] R.H. Abraham and C.D. Shaw, *Dynamics: The Geometry of Behavior*, parts 1-4, Ariel Press, Santa Cruz, CA. (1984).

[12] M.J. Lighthill, “Attachment and Separation in Three Dimensional Flow,” *Laminar Boundary Layers II*, ed. L. Rosenhead, pp. 72-82, Oxford University Press (1963).

[13] A.E. Perry and B.D. Fairly, “Critical Points in Flow Patterns,” *Advances in Geophysics* 18 B pp. 299-315 (1974).

[14] M.S. Chong. A.E. Perry, B.J. Cantwell, “A General Classification of Three-Dimensional Flow Fields,” *Phys. Fluids* A **2** (5) pp. 765-777, May (1980).

[15] A.E. Perry, “A Study of Degenerate and Non-degenerate Critical Points in Three-dimensional Flow Fields,” Deutsche Forschungs- und Versuchsanstalt fur Luft und Raumsfahrt report DFVLR-FB 84-36.

[16] U. Dallman and G. Schewe, “On the Topological Changes of Separating Flow Structures at Transition Reynolds Numbers,” American Institute of Aeronautics and Astronautics, paper AIAA-87-1266.

[17] U. Dallman, “Three-dimensional Vortex Structures and Vorticity Topology,” *Fluid Dynamics Research* **3** pp. 183-189 (1988).

[18] A.E. Perry and J.H. Watmuff, “The Phase-Averaged Large-Scale Structures in Three-Dimensional Turbulent Wakes,” *J. Fluid Mech*. **103** pp. 33-51. A.E. Perry and D.K. Tan, “Simple Three-Dimensional Vortex Motions in Coflowing Jets and Wakes,” *J. Fluid Mech*. **141** pp. 197-231, (1984).

[19] S. Shirayama and K. Kuwahara, “Flow Past a Sphere: Topological Transitions of the Vorticity Field,” American Institute of Aeronautics and Astronautics, paper AIAA-90-3105-CP.

[20] L. Hesselink and J. Helman, “Evaluation of Flow Topology from Numerical Data,” American Institute of Aeronautics and Astronautics, paper AIAA-87-1181.

[21] J. L. Helman and L. Hesselink, “Surface Representation of Two- and Three-Dimensional Fluid Flow Topology,” *Proc. Visualization ‘90,* San Francisco, IEEE Computer Society Press. (1990).

[22] J. L. Helman and L. Hesselink, “Analysis and Representation of Complex Structures in Separated Flows,” S*PIE Conf on Extracting Meaning From Complex Data,* San Jose, (1991).

[23] J.L. Helman and L. Hesselink, “Representation and Display of Vector Field Topology in Fluid Flow Data Sets,” *IEEE Computer,* pp. 27-36, Aug. 1989. Also appears in *Visualization in Scientific Computing*, G. M. Fielson & B. Shriver, eds. Companion videotape available from IEEE Computer Society Press.

[24] A. Globus, “Octree Optimization,” SPIE paper 1459-01, *SPIE Conf on Extracting Meaning From Complex Data*, San Jose (1991).

[25] Mike Yamasaki, NASA Ames Research Center, personal communication.

[26] P. P. Walatka, P. G. Buning, *PLOT3D User’s Manual,* NASA Technical Memorandum 101067, NASA Ames Research Center.

[27] J.A. Benek, P.G. Buning, J.L. Steger, “A 3-D Chimera Grid Embedding Technique,” *AIAA 7th Computational Fluid Dynamics Conference*, Cincinnati, Ohio (1985), AIAA-85-1523.

[28] R.A. Walker, “Computing the Jordan Form for Control of Dynamic Systems,” Guidance and Control Laboratory, Department of Aeronautics and Astronautics, Stanford University (1981).

[29] W.H. Press, et al., *Numerical Recipes in C: The Art of Scientific Computing,* Cambridge University Press, Cambridge (1988).

[30] C.H. Hung, P.G. Buning, “Simulation of Blunt-Fin-Induced Shock-Wave and Turbulent Boundary-Layer Interaction,” *J. Fluid Mech.* (1985), Col. 154, pp. 163-185.

[31] Y.M. Rizk, S. Ben-Shmuel, “Computation of the Viscous Flow Around the Shuttle Orbiter at Low Supersonic Speeds,” *AIAA 23rd Aerospace Sciences Meeting*, Jan. 14-17, (1985) Reno, Nevada, AIAA-85-0168.

[32] S. Rogers, D. Kwak, an U. Kaul, “A Numerical Study of Three-Dimensional Incompressible Flow Around Multiple Posts,” American Institute of Aeronautics and Astronautics, paper AIAA-86-0353.

[33] P.G. Buning, I.T. Chiu, S. Obayashi, Y.M. Rizk, and J.L. Steger, “Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent,” *Proc. AIAA Atmospheric Flight Mechanics Conference*, American Institute of Aeronautics and Astronautics, Minneapolis (1988).

[34] S. Ying, L.Schiff and J.L. Steger “A Numerical Study of Three-Dimensional Separated Flow Past a Hemisphere Cylinder” *Proc. AIAA 19th Fluid Dynamics, Plasma Dynamics and Lasers Conference.*

[35] L.A. Yates and G.T. Chapman, “Streamlines, Vorticity Lines, and Vortices,” American Institute of Aeronautics and Astronautics, paper AIAA-91-0731.

[36] A.E. Perry and H. Hornung, “Some Aspects of Three-Dimensional Separation, Part II: Vortex Skeletons,” *Z. Flugwiss, Weltraumforsch*. **8**, Heft 3, pp. 155-160 (1984).

[37] A. Globus, C. Levit, T. Lasinski, “A Tool for Visualizing the Topology of Three-Dimensional Vector Fields,” Report RNR-91-017, Applied Research Branch, MS T045-1, NASA Ames Research Center, Moffett Field, CA, 94035.